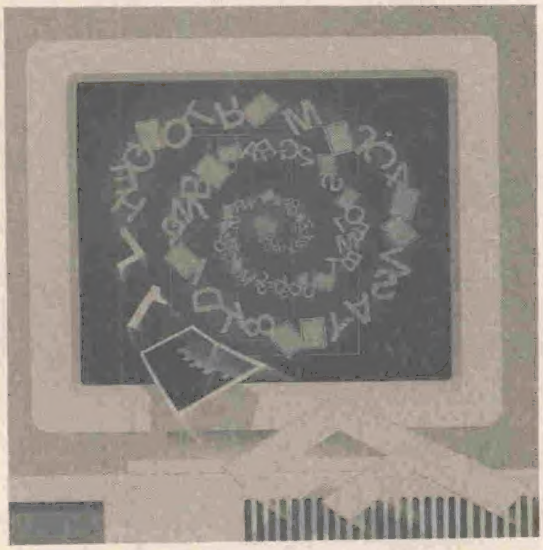
ปราโมทย์ พลังสันติกุล

SOFT MAIL

เขียนโปรแกรมติดต่อ I/O Port ด้วย Visual Basic (ตอนที่ 1)



การใช้งานโปรแกรม Visual Basic ในส่วนของการติดต่อกับพอร์ตอินพุตและเอาต์พุต

จากบทความตอนที่แล้วที่ผู้เขียนได้เขียนเรื่องเกี่ยวกับ Communication programming in Visual Basic : Communication custom control (MSComm) ซึ่งเป็นการใช้งาน Visual Basic ที่เกี่ยวข้องกับการติดต่อผ่านทาง RS-232 (Serial Communication) ทำให้เราสามารถเขียนโปรแกรมโดยใช้ Visual Basic ติดต่อกับอุปกรณ์ต่าง ๆ ผ่านทาง RS-232 ได้นั้น แต่ก็ยังขาดความสมบูรณ์ในแง่ที่หากผู้ใช้ต้องการติดต่อกับอุปกรณ์ภายนอกโดยไม่ได้ผ่านทาง RS-232 (Serial Communication) แต่ต้องการที่จะติดต่อผ่านทางพอร์ตขนาน (Parallel Port) หรือการ์ดควบคุมต่าง ๆ ที่มีพอร์ตมาให้ เช่น Card I/O 8255 ต่าง ๆ เป็นต้น เพื่อเป็นการขยายระบบอินพุตและเอาต์พุตให้ใช้งานได้มากยิ่งขึ้น และเป็นการติดต่อกับอุปกรณ์ดังกล่าวผ่านทางพอร์ตโดยตรง ซึ่งสามารถทำให้การควบคุมอุปกรณ์ต่าง ๆ ทำได้มากขึ้น นอกเหนือจากทาง RS-232 เท่านั้น โดยการ์ดควบคุมดังกล่าวนั้นนิยมใช้การ์ดที่มี IC8255 (Programmable Pheripheral Interface) มาใช้เป็นพอร์ตควบคุม ซึ่งเป็นพอร์ตข้อมูลแบบขนาน

เมื่อผู้เขียนมีความคิดดังกล่าว จึงได้ค้นคว้าว่า Visual Basic มีความสามารถทางด้านนี้หรือไม่ ปรากฏว่าความสามารถในการติดต่อกับพอร์ตโดยตรงของ Visual Basic นั้นไม่มีหรือไม่สามารถทำได้ ทำให้ผู้เขียนมองไปที่ API ที่มีอยู่ของวินโดวส์ หรือที่เรารู้จักกันดีที่เรียกว่า WIN32 API ซึ่งเป็นการใช้งาน API ผ่านทาง Visual Basic ปรากฏว่าผู้เขียนก็ไม่พบเช่นกัน และหากไปค้นดูในหนังสือเกี่ยวกับ Visual Basic Programmer's Guide To The WIN32 API โดยละเอียดจะพบข้อมูลเกี่ยวกับการติดต่อกับพอร์ตโดยตรง แต่ไม่ใช่ของ WIN32 API แต่กลับเป็นการเขียน Dynamic-Link Library (DLL) ขึ้นมาซึ่งของผู้แต่งหนังสือเล่มนั้น หากผู้อ่านไม่มีหนังสือเล่มดังกล่าวก็จะไม่สามารถที่จะติดต่อกับพอร์ตได้โดยตรง เพราะฉะนั้นมีทางเดียวที่จะใช้งานพอร์ตได้โดยตรงคือการเขียน Dynamic-Link Library (DLL) ขึ้นมาใช้งานเอง

ในการเขียน Dynamic-Link Library (DLL) นั้นเราก็สามารถใช้โปรแกรมที่สามารถใช้ในการเขียน DLL ได้คือ Pascal, Delphi, Borland C++, C++ Builder, Visual C++ ฯลฯ ฉะนั้นบทความนี้ผู้เขียนจะนำเสนอออกเป็น 2 ส่วน คือ การเขียน DLL ที่เกี่ยวข้องกับการติดต่อพอร์ต โดยผู้เขียนเลือกใช้ Visual C++ ในการพัฒนา และอีกส่วนหนึ่งจะเป็นการเขียน Visual Basic เพื่อเรียกใช้งานในการติดต่อกับพอร์ตผ่านทาง DLL ที่เราจะพัฒนาใช้ขึ้นมาเอง ซึ่งในส่วนของการเขียน DLL ด้วย Visual C++ นั้นผู้เขียนจะไม่อธิบายรายละเอียดไว้ แต่เป็นหน้าที่ของผู้อ่านที่จะต้องไปทำความเข้าใจ จะอธิบายส่วนของ Visual Basic ในการใช้งาน DLL เท่านั้น เพราะผู้อ่านบางท่านอาจจะไม่ได้ศึกษาภาษา C มาก่อน จึงจะไม่อธิบายรายละเอียดไว้

VBPORT.DLL

Dynamic-Link Library (DLL)

ในการเขียน Dynamic-Link Library (DLL) ใน WIN32 จะต้องมีการเปรียบเทียบตัวแปรระหว่างภาษา C กับ Visual Basic เพื่อที่จะผ่านค่าไปให้ฟังก์ชันภาษา C กับ Visual Basic ได้ถูกต้อง ผู้เขียนจะเปรียบเทียบตัวแปรเฉพาะที่ต้องใช้เท่านั้น ซึ่งมีดังนี้

การเปรียบเทียบค่าตัวแปรของ C/C++, C/C++ (WIN32) และ VB (Win32)

C/C++	C/C++(WIN32)	VB(Win32)
int	BOOL	Long
unsigned shot	WORD	Integer
unsigned long	DWORD	Long

Dynamic-Link Library (DLL) ที่ได้ทำการพัฒนาขึ้นมานั้น จะประกอบไปด้วยคำสั่งทั้งส่วนของ INPUT และ OUTPUT PORT โดยมีฟังก์ชันดังต่อไปนี้คือ

1. ส่วนของ INPUT PORT ฟังก์ชัน vbinportp, vbinportw, vbinportd เป็นฟังก์ชันในการอ่านค่าจาก INPUT PORT ซึ่งค่าของ PORT นั้นจะมีค่าอยู่ระหว่าง 0 – 65,535 โดยฟังก์ชันประกอบไปด้วย

- **vbinportp** เป็นการอ่านค่าของจำนวน BYTE
- **vbinportw** เป็นการอ่านค่าของจำนวน WORD
- **vbinportd** เป็นการอ่านค่าของจำนวน DOUBLE WORD

Input a byte (_inp) from a port.

ใน Visual Basic ประกาศค่าใน Moudle (file.BAS)

Declare Function vbinportp LIB "VBPORT"
(ByVal port As Integer) As Long

เทียบได้กับภาษา C/C++ ดังนี้

```
BOOL vbinportp( WORD port )
{
return(_inp( port ));
}
```

Input a word (_inpw) from a port.

ใน Visual Basic ประกาศค่าใน Moudle (file.BAS)

Declare Function vbinportw LIB "VBPORT"
(ByVal port As Integer) As Integer

เทียบได้กับภาษา C/C++ ดังนี้

```
WORD vbinportw( WORD port )
{
return(_inpw( port ));
}
```

Input a double word (_inpd) from a port.

ใน Visual Basic ประกาศค่าใน Moudle (file.BAS)

Declare Function vbinportd LIB "VBPORT"
(ByVal port As Integer) As Long

เทียบได้กับภาษา C/C++ ดังนี้

```
DWORD vbinportd( WORD port )
{
return(_inpd( port ));
}
```

2. ส่วนของ OUTPUT PORT ฟังก์ชัน vboutportp, vboutportw, vboutportd เป็นฟังก์ชันในการเขียนค่าไปที่ OUTPUT PORT ซึ่งค่าของ PORT นั้นจะมีค่าอยู่ระหว่าง 0 – 65,535 โดยฟังก์ชันประกอบไปด้วย

- **vboutportp** เป็นการส่งค่าของจำนวน BYTE
- **vboutportw** เป็นการส่งค่าของจำนวน WORD
- **vboutportd** เป็นการส่งค่าของจำนวน DOUBLE WORD

Output a byte(_outp) at a port.

ใน Visual Basic ประกาศค่าใน Moudle (file.BAS)

Declare Function vboutportp LIB "VBPORT"
(ByVal port As Integer , ByVal databyte As Long) As Long

เทียบได้กับภาษา C/C++ ดังนี้

```
BOOL APIENTRY vboutportp(WORD port,
BOOL databyte)
{
return(_outp(port, databyte ));
}
```

Output a word(_outpw) at a port.

ใน Visual Basic ประกาศค่าใน Moudle (file.BAS)

Declare Function vboutportw LIB "VBPORT"
(ByVal port As Integer , ByVal dataword As Integer) As Integer

เทียบได้กับภาษา C/C++ ดังนี้

```
WORD APIENTRY vboutportw(WORD port,
WORD dataword)
{
return(_outpw(port, dataword));
}
```

Output a double word (_outpd) at a port.

ใน Visual Basic ประกาศค่าใน Moudle (file.BAS)

Declare Function vboutportd LIB "VBPORT"
(ByVal port As Integer , ByVal dataword As Long) As Integer

เทียบได้กับภาษา C/C++ ดังนี้

```
DWORD APIENTRY vboutportd(WORD port,
DWORD dataword)
{
return(_outpd(port, dataword));
}
```

เขียนโปรแกรมติดต่อ I/O Port ด้วย Visual Basic
(ตอนที่ 1)

ในส่วนของการเขียนโปรแกรมด้วย Visual Basic เพื่อใช้งานเกี่ยวกับ Input/Output Port นั้น ผู้เขียนได้ใช้ Card I/O 8255 ในการทดสอบการใช้งาน Dynamic-Link Library (DLL) ดังกล่าว

IC8255 เป็นไอซีซึ่งสามารถทำหน้าที่ได้ทั้งพอร์ตอินพุตและเอาต์พุตโดยการโปรแกรมควบคุมด้วยซอฟต์แวร์ ซึ่งมีคุณสมบัติดังนี้ คือ

1. มีอินพุตและเอาต์พุต 24 ทาง ที่สามารถโปรแกรมได้
2. มีอินพุตและเอาต์พุตทั้งหมด 3 พอร์ต คือ พอร์ต A, พอร์ต B, พอร์ต C พอร์ตละ 8 บิต
3. โปรแกรมให้เป็นอินพุตหรือเอาต์พุตได้ กำหนดโหมดการทำงานได้ 3 โหมด

ซึ่งรายละเอียดของ IC8255 นั้นผู้อ่านสามารถหารายละเอียดได้จากหนังสือเกี่ยวกับ IC ได้ทั่วไป หรือจากหนังสือที่เกี่ยวข้องกับอุปกรณ์ IC, หนังสืออิเล็กทรอนิกส์ทั่วไป

อธิบายโปรแกรมการทำงานของโพรซีเดอร์

ในส่วนของโปรแกรม Visual Basic ในการใช้งานเพื่อติดต่อกับ Card I/O 8255 มีดังนี้ คือ

MODVBPORT.BAS ชื่อ Module modVBPort

การประกาศฟังก์ชัน API ซึ่งจะต้องประกาศไว้ใน Module ของ Visual Basic ดังนี้คือ

```
ส่วนของการเอาต์พุตพอร์ต
- Declare Function vboutportp Lib "VBPORT"
(ByVal port As Integer, ByVal databyte As Long) As Long
- Declare Function vboutportw Lib "VBPORT"
(ByVal port As Integer, ByVal dataword As Integer) As Integer
- Declare Function vboutportd Lib "VBPORT"
(ByVal port As Integer, ByVal dataword As Long) As Integer
ส่วนของการอินพุตพอร์ต
- Declare Function vbinportp Lib "VBPORT"
(ByVal port As Integer) As Long
- Declare Function vbinportw Lib "VBPORT"
(ByVal port As Integer) As Integer
- Declare Function vbinportd Lib "VBPORT"
(ByVal port As Integer) As Long
```

FRMINOUTPORT.FRM ชื่อ FORM FRMINOUTPORT

การประกาศตัวแปรใช้ใน FORM ดังนี้ คือ

```
Dim dataPortA, dataPortB, dataPortC
Dim PORTA, PORTB, PORTC
Const PORTCW = &H303
Const MODE_CONTROL_8255 = &H80
```

Sub ที่กำหนดใน frmInOutPort.Frm ชื่อ frmInOutPort

Private Sub Form_Load()

โพรซีเดอร์ในส่วน Form Load เป็นการกำหนดค่า I/O Port 8255 นั้นให้เป็นการเอาต์พุตทุกพอร์ต โดยการกำหนดค่าดังนี้ คือ

```
Call vboutportp(PORTCW, MODE_CONTROL_8255) '— Port Control
Text1.Text = "300"          ' ADDRESS PORT A = &H300 (Hex)
Text2.Text = "301"          ' ADDRESS PORT B = &H301 (Hex)
Text3.Text = "302"          ' ADDRESS PORT C = &H302 (Hex)
Text4.Text = "FF"           ' Data out to port A
Text5.Text = "55"           ' Data out to port B
Text6.Text = "0F"           ' Data out to port C
```

PORTCW = &H303 ซึ่งเป็น Address ของ PORT CONTROL ของ I/O 8255

MODE_CONTROL_8255 = &H80 เป็นการกำหนดค่าให้ PORT CONTROL ให้มีค่าเท่ากับ &H80 ซึ่งเป็นโหมดการทำงานให้ทุกพอร์ตเป็นการเอาต์พุตทุกพอร์ต

ในการใช้งาน PORT I/O 8255 เราจะต้องกำหนดโหมดการทำงานของ พอร์ต A, B, C ก่อนการใช้งานว่าต้องการให้พอร์ตทั้งหมดทำงานในรูปแบบไหน

หมายเหตุ ค่าต่าง ๆ ที่กำหนดให้กับ I/O Port 8255 สามารถศึกษารายละเอียดได้จากเอกสารของ IC8255 และ CARD I/O 8255 นั้น ๆ อุปกรณ์ CARD I/O8255 หาซื้อได้โดยค้นจากหนังสือเซมิฯ อิเล็กทรอนิกส์กำหนดค่า Address ของ CARD I/O 8255 คือ

&H300 = PORT A ของ 8255
&H301 = PORT B ของ 8255
&H302 = PORT C ของ 8255
&H303 = PORT CONTROL ของ 8255

Private Sub Command1_Click()

โพรซีเดอร์ในส่วน Command1_Click เป็นการกำหนดค่า I/O Port 8255 ของพอร์ต A ในการเอาต์พุตค่า

```
PORTA = Val("&H" & Text1.Text)
dataPortA = Val("&H" & Text4.Text)
Call vboutportp(PORTA, dataPortA) '- Port A
PORTA = &H300 เป็นตำแหน่ง Address ของ PORT A
dataPortA เป็นค่าที่เอาต์พุต ไปที่ PORT A จากตัวอย่างส่งค่าไป
เท่ากับ &HFF = 11111111 (ฐาน2) คือให้ ค่า 8 BIT นั้นมีค่าทั้งหมด
หรือ LED ติดทุกดวง
```

Private Sub Command2_Click()

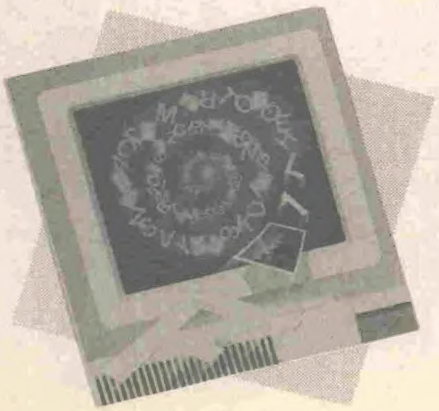
โพรซีเดอร์ในส่วน Command1_Click เป็นการกำหนดค่า I/O Port 8255 ของพอร์ต B ในการเอาต์พุตค่า

```
PORTB = Val("&H" & Text2.Text)
dataPortB = Val("&H" & Text5.Text)
Call vboutportp(PORTB, dataPortB) '- Port B
PORTB = &H301 เป็นตำแหน่ง Address ของ PORT B
dataPortB เป็นค่าที่เอาต์พุต ไปที่ PORT B จากตัวอย่างส่งค่าไป
เท่ากับ &H55 = 10101010 (ฐาน2)คือให้ ค่า 8 BIT
นั้นมีค่าสลับกัน หรือ LED ติดดับสลับกัน
```

Private Sub Command3_Click()

โพรซีเดอร์ในส่วน Command3_Click เป็นการกำหนดค่า I/O Port 8255 ของพอร์ต C ในการเอาต์พุตค่า

```
PORTC = Val("&H" & Text3.Text)
dataPortC = Val("&H" & Text6.Text)
Call vboutportp(PORTC, dataPortC) '- Port C
PORTC = &H302 เป็นตำแหน่ง Address ของ PORT C
dataPortC เป็นค่าที่เอาต์พุต ไปที่ PORT C จากตัวอย่างส่งค่าไป
เท่ากับ &H0F = 00001111 (ฐาน2)คือให้ ค่า 8 BIT นั้นมีค่าสลับกัน
อย่างละ 4 BIT หรือ LED ติด 4 ดวง ดับ 4 ดวง
```



Private Sub Command4_Click()

โพรซีเดอร์ในส่วน Command4_Click เป็นการกำหนดค่า I/O Port 8255 พอร์ต A, B, C และให้เอาต์พุตค่าเพื่อเคลียร์ค่าทั้งหมด

```
Call vboutportp(PORTA, 0) '- Port A
Call vboutportp(PORTB, 0) '- Port B
Call vboutportp(PORTC, 0) '- Port C
ค่า 0 เป็นการกำหนดค่าให้ I/O Port 8255
เป็นเอาต์พุตค่าเพื่อเคลียร์ค่าทั้งหมด คือ 0 = 00000000 (ฐาน2)
คือให้ LED ดับหมด
Private Sub
Command6_Click(),Command7_Click(),
Command8_Click()
โพรซีเดอร์ในส่วน Command6_Click, Command7_Click,
Command8_Click เป็นการอ่านค่า I/O Port 8255 พอร์ต A,B,C
Private Sub Command6_Click() คือ
PORTA = Val("&H" & Text1.Text)
Label4.Caption = Hex(vbinportp(PORTA))
Private Sub Command7_Click() คือ
PORTB = Val("&H" & Text2.Text)
Label5.Caption = Hex(vbinportp(PORTB))
Private Sub Command8_Click() คือ
PORTC = Val("&H" & Text3.Text)
Label6.Caption = Hex(vbinportp(PORTC))
ซึ่งเป็นการอ่านค่าจาก พอร์ต A, B, C ตามลำดับ
```

การทดสอบและใช้งานโปรแกรม VBPort

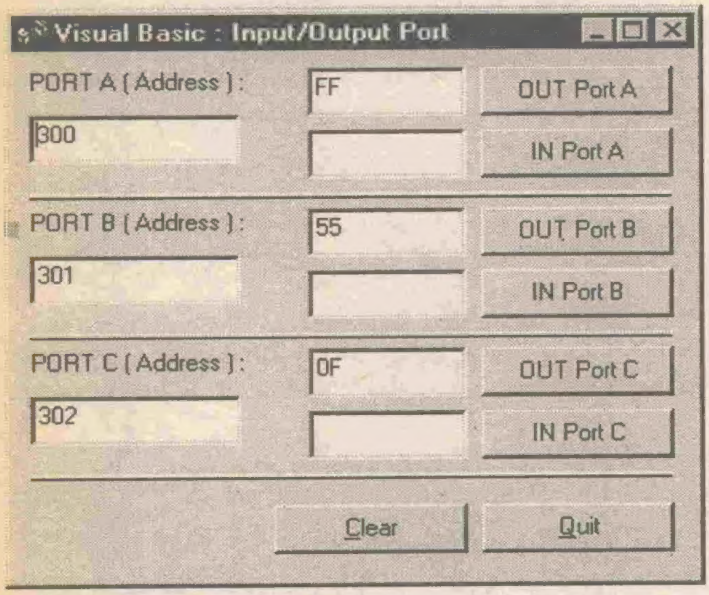
โปรแกรมตัวอย่าง VBPort นั้นจะมีหน้าตาดังรูปที่ 1 ซึ่งจะต้องมีอุปกรณ์ในการใช้งานเพื่อทดสอบการทำงานของโปรแกรมด้วย คือ

1. Card I/O 8255 ในการทดสอบ พอร์ต A, B, C
2. ปุ่มเอาต์พุตพอร์ต ทั้งพอร์ต A, B, C และปุ่มเคลียร์ค่า
3. ปุ่มอินพุตพอร์ต ทั้งพอร์ต A, B, C
4. ปุ่ม Quit เป็นการเลิกการทำงานของโปรแกรม
5. ป้อนค่าที่ต้องการให้พอร์ตทำการส่ง RUN โปรแกรมทดสอบได้เลยครับ

ในการส่งค่าให้กับพอร์ตนั้นจะทำการส่งครั้งละ 8 บิต ต่อ 1 พอร์ต คือ กำหนดค่าส่งได้ดังนี้ หากต้องการส่งค่าที่ตำแหน่ง 1, 3, 4, 7 จะกำหนดเป็นเลขฐานสองได้ดังนี้ 01011001 = &H59 จากโปรแกรมตัวอย่างเป็นการส่งค่า &H55 = 10101010 (ฐาน2) คือเป็นการส่งค่าในตำแหน่งที่ 0, 2, 4, 6 โดยเราจะส่งค่าให้กับพอร์ตในค่าเลขฐานสิบหก ซึ่งการแปลงเลขค่าฐานต่าง ๆ ผู้อ่านสามารถใช้โปรแกรม Calculator ที่มากับ WINDOWS ได้โดยทำงานในโหมด SCIENTIFIC ครับ

ประโยชน์ที่ได้รับจากบทความ

1. เรียนรู้วิธีการเขียน Dynamic-Link Library (DLL) โดยการศึกษาจากโปรแกรมตัวอย่าง
2. เรียนรู้ถึงการใช้งาน Visual Basic ในการติดต่อกับ API VBPORT.DLL นอกเหนือจาก WIN32 API



รูปที่ 1 โปรแกรม VBPort

3. ได้แนวความคิดหากส่วนไหนที่ Visual Basic ไม่สามารถทำได้ สามารถพัฒนา Dynamic-Link Library (DLL) ขึ้นมาใช้แทนได้

4. ใช้งาน Visual Basic ในการติดต่อกับอุปกรณ์ภายนอกโดยการผ่านทาง PORT I/O นอกเหนือจากพอร์ต RS-232

ขอบเขตของ VBPORT.DLL Dynamic-Link Library (DLL)

VBPORT.DLL ผู้เขียนได้พัฒนาโดยใช้ Visual C++ ซึ่งจะทำงานบนระบบวินโดวส์ 95 ส่วนวินโดวส์ 98 นั้นผู้เขียนไม่ได้ลงระบบดังกล่าวไว้จึงไม่ได้ทดสอบว่าจะใช้งานได้หรือไม่ ส่วนบนระบบวินโดวส์ 3.1 นั้นไม่สามารถใช้งานได้ แต่หากผู้อ่านสนใจและต้องการให้ผู้เขียนนำเสนอในส่วนของวินโดวส์ 3.1 ผู้เขียนจะนำมาเสนอในครั้งต่อไป

สรุปเนื้อหา

หลังจากได้ศึกษาการใช้งานโปรแกรมตัวอย่าง ซึ่งจะเป็นการส่งค่าไปที่ตำแหน่งของพอร์ตนั้นและทำการอ่านค่ากลับมา หากไม่มีอุปกรณ์ CARD I/O 8255 ผู้อ่านลองกำหนดตำแหน่งใหม่ให้กับ PORT A, B, C ไปที่ตำแหน่ง 0 - 6, 9 – 12, 14 – 22 ตำแหน่งใดตำแหน่งหนึ่ง แล้วส่งค่าไปและอ่านค่ากลับมา เพื่อเป็นการทดสอบการใช้งาน DLL กับ Visual Basic ซึ่งตำแหน่งดังกล่าวผู้เขียนได้ทดสอบแล้ว เช่น กำหนด Const PORTA = &H0 , Const PORTB = &H1 , Const PORTC = &H2 เป็นต้น ผู้เขียนก็หวังว่าผู้เริ่มต้นการใช้งานโปรแกรม Visual Basic ในส่วนของการติดต่อกับพอร์ตอินพุตและเอาต์พุต คงพอจะเป็นแนวทางในการพัฒนาโปรแกรมเพื่อนำไปประยุกต์ใช้งานทางด้านติดต่อกับอุปกรณ์ภายนอก หรือเพื่อใช้ในการควบคุมอุปกรณ์ต่าง ๆ ผ่านทางพอร์ตดังกล่าว ซึ่งจะทำให้มีความสมบูรณ์มากยิ่งขึ้น หลังจากที่ได้ศึกษาการติดต่อผ่านทาง RS-232 จากฉบับที่ผ่านมาแล้ว ส่วนในตอนต่อไปผู้เขียนจะนำเสนอเกี่ยวกับการติดต่อกับพอร์ตขนานของเครื่องพิมพ์ ซึ่งจะแสดงรายละเอียดของพอร์ตขนานของเครื่องพิมพ์ สร้างอุปกรณ์ทดสอบ พร้อมกับโปรแกรมทดสอบการใช้งาน

ท้ายบทความ

การใช้ Visual Basic สำหรับการเขียนโปรแกรมติดต่อกับพอร์ตโดยตรง ถึงแม้ว่า Visual Baisc จะไม่สามารถทำได้ แต่เราก็สามารถใช้โปรแกรมอื่นในการนำมาประยุกต์ เพื่อให้ Visual Basic เรียกใช้งานได้อย่างไม่มีปัญหา ขึ้นอยู่กับการประยุกต์ใช้งานและการเลือกเครื่องมือในการพัฒนางานของเราเป็นสำคัญ โดยเฉพาะการใช้งานในด้านการควบคุมอุปกรณ์ต่าง ๆ นับว่ามีความสำคัญไม่น้อยนอกเหนือจากการพัฒนางานด้วยโปรแกรมล้วน ๆ ซึ่งผู้เขียนหวังว่าคงจะพอเป็นแนวทางให้ผู้อ่านนำไปประยุกต์กับการใช้งานในด้านต่าง ๆ โดยเฉพาะยุคเศรษฐกิจที่เรียกว่า ยุค IMF คงจะต้องพึ่งตนเองเป็นอย่างมาก "ทุกอย่างมีหนทางแก้ไข เพียงแต่เราลองมองในอีกด้านหนึ่งดูบ้าง แล้วจะหาหนทางได้" สวัสดีครับ

หมายเหตุ ผู้ที่สนใจซอร์สโค้ดของโปรแกรม VBPort สามารถใช้บริการซอฟต์เมล์ได้ครับ